

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-234978

(43)Date of publication of application : 13.09.1996

(51)Int.Cl.

G06F 9/38

G06F 9/38

(21)Application number : 07-036525

(71)Applicant : FUJITSU LTD

(22)Date of filing : 24.02.1995

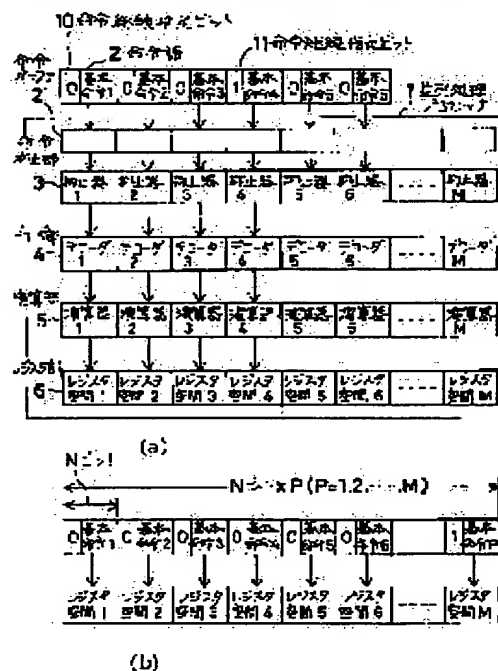
(72)Inventor : NAKAJIMA YASUHIKO
TAKEBE YOSHIMASA

(54) PARALLEL PROCESSOR AND MULTI-INSULATION PARALLEL EXECUTION METHOD

(57)Abstract:

PURPOSE: To effectively process the instructions with use of a limited number of registers by preventing the input of the basic instructions following the specific one that is inhibited to be processed in the same timing as others and regarding the basic instructions which precede the specific one and can be calculated in the same timing as a single instruction word to calculate these instructions in parallel to each other.

CONSTITUTION: Each of suppressors 1 to M decides the value of the instruction continuation suppression bit of every input basic instruction. If this bit value is equal to 0, the instructions that are processed in the same timing and in parallel to each other are decided. If the bit value is equal to 1, the subsequent instructions are suppressed to be inputted to a decoder 4. Then a range covering a basic instruction row where the instruction continuation bits of value 0 are continuous through a basic instruction of an instruction continuation bit of value 1 emerging after a series of instruction continuation bits is regarded as a single instruction word. These instruction words can be processed in the same timing up to M pieces. That is, the length of single instruction words that can be processed in parallel to each other can include M pairs of them when the numbers P of basic instructions are set at 1, 2...M.



LEGAL STATUS

[Date of request for examination] 19.02.2001

[Date of sending the examiner's decision of rejection] 03.02.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3569338

[Date of registration] 25.06.2004

[Number of appeal against examiner's decision of rejection] 2004-04343

[Date of requesting appeal against examiner's decision of rejection] 04.03.2004

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-234978

(43) 公開日 平成8年(1996)9月13日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/38	3 1 0		G 0 6 F 9/38	3 1 0 X
	3 7 0			3 7 0 X

審査請求 未請求 請求項の数 5 O L (全 16 頁)

(21) 出願番号 特願平7-36525

(22) 出願日 平成7年(1995)2月24日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 中島 康彦

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72) 発明者 竹部 好正

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74) 代理人 弁理士 長谷川 文廣 (外2名)

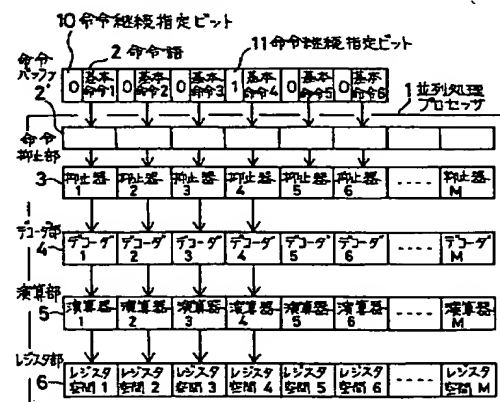
(54) 【発明の名称】 並列処理プロセッサと複数命令の並列実行方法

(57) 【要約】

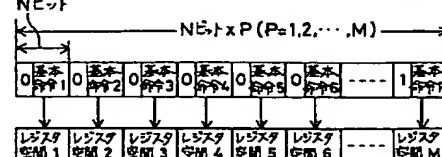
【目的】 並列処理プロセッサと複数命令の並列実行方法に関し、命令語から並列処理が可能な一連の基本命令を認識し、可変長の一命令として扱うことにより効率よく並列処理することを目的とする。

【構成】 基本命令は同一タイミングで処理することの可否を指定する命令継続指定ビットを設けたものとし、命令バッファの基本命令をそれぞれに演算する演算部と、それぞれの演算部の演算結果を格納するレジスタ、命令継続ビットの値を判別して基本命令のうち同一タイミングで処理を行ってはいならない基本命令の演算部への入力を抑止する命令抑止部とを備え、同一タイミングで処理をしてはいならないことを指示された基本命令より後の基本命令の演算部への入力は抑止し、それ以前の同一タイミングで演算できる引き続き基本命令を一命令語とみなし、同一タイミングで演算部に入力して並列処理する構成を持つ。

本発明の基本構成



(a) 本発明の並列プロセッサの基本構成



(b) 本発明の命令実行方法

【特許請求の範囲】

【請求項 1】 複数の基本命令を一命令語としてまとめて複数の基本命令を並列処理する並列処理プロセッサにおいて、基本命令は同一タイミングで処理することの可否を指定する命令継続指定ビットを設けたものとし、複数の基本命令を基本命令毎に格納する命令バッファと、該命令バッファの基本命令をそれぞれに演算する演算部と、それぞれの演算部の演算結果を格納するレジスタ、該命令継続ビットの値を判別して基本命令のうち同一タイミングで処理を行ってはならない基本命令の演算部への入力を抑止する命令抑止部とを備え、該命令抑止部は同一タイミングで処理をしてはならないことを指示された基本命令より後の基本命令の演算部への入力は抑止し、それ以前の同一タイミングで演算できる引き続く基本命令を一命令語とみなし、同一タイミングで演算部に入力して並列処理することを特徴する並列処理プロセッサ。

【請求項 2】 複数の基本命令を一命令語としてまとめて複数の基本命令を並列処理する並列処理プロセッサにおける複数命令の並列実行方法において、基本命令に同一タイミングで演算することの可否を指定する命令継続指定ビットを持たせ、複数の基本命令の各々を入力して演算する演算部と、該命令継続ビットの値を判別して基本命令のうち同一タイミングで演算を行ってはならない基本命令の演算を抑止する命令抑止部とを備え、該命令抑止部は同一タイミングで演算をしてはならないことを指示された基本命令より後の基本命令の演算部への入力は抑止し、それ以前の同一タイミングで演算できる引き続く基本命令を一命令語とみなし、該一命令を同一タイミングで演算部に入力して並列処理することを特徴する複数命令の並列実行方法。

【請求項 3】 Nビットの基本命令を処理する複数のレジスタにより構成されるレジスタ群を 1 レジスタ空間とした時、一命令に含まれる各基本命令はそれぞれ異なるレジスタ空間で扱われるものであり、使用できるレジスタ空間数が M 個であるとき、一命令として扱う基本命令数は最大 M であり、1 個から最大 M 個までのレジスタ空間を使用して命令を実行することを特徴とする請求項 2 に記載の複数命令の並列実行方法。

【請求項 4】 レジスタ空間 A のデータを格納するバッファレジスタ a とレジスタ空間 B のデータを格納するバッファレジスタ b を設け、バッファレジスタ a とバッファレジスタ b は接続されてバッファレジスタ a のデータはバッファレジスタ b に複写されるものであり、レジスタ空間 A からレジスタ空間 B にデータを送信する場合に、レジスタ空間 A に対応する基本命令に対してレジスタ空間 A のデータをバッファレジスタ a にストアする命令 X を記述し、レジスタ空間 B に対応する基本命令にバッファレジスタ b からレジスタ空間 B にロードする命令 Y を記述し、命令 X を処理するタイミングでバッファレ

ジスタ a にレジスタ空間 A のデータをストアし、それより後のタイミングで命令 Y を実行してバッファレジスタ a からバッファレジスタ b に転送されたデータをレジスタ空間 B にロードすることによりレジスタ空間の間でデータ転送を行うことを特徴とする請求項 3 に記載の複数命令の並列実行方法。

【請求項 5】 一命令語に M 個の基本命令が含まれるとした時、M ビットからなる命令拡張制御レジスタと、M 個の基本命令のうち有効な基本命令の位置を表すマスクパターンを該命令拡張制御レジスタにセットし、該命令拡張制御レジスタを参照して命令語を生成する命令語生成部とを備え、入力された基本命令 I に対して、該マスクパターンを参照し、マスクパターンの有効な基本命令の位置に該基本命令 I をセットし、有効でないことを示すマスクパターンに対応する位置に無操作命令もしくはなにも命令を与えないことにより M 個の基本命令をもつ命令語を生成することを特徴とする請求項 2 に記載の複数命令の並列実行方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数の基本命令を一命令として並列処理する並列処理プロセッサと複数命令の並列実行方法に関する。

【0002】計算機システムでは、パイプライン処理等で複数命令を並列処理し、高速化を計っている。従来は複数命令をまとめて固定長の一命令にし、複数命令を並列処理する長形式命令語方式がとられているが、可変長でないため、基本命令の数が少ない場合にはキャッシュメモリに保持したときのヒット率の低下をまねいていた。また、スーパースカラ方式等で命令並列度に応じて並列処理することも行われているが、並列度が大きくなるのに応じてレジスタ数を増大させる必要がある。

【0003】本発明は、並列処理が可能な一連の基本命令を認識し、それを可変長の一命令として扱うことにより効率よく命令の並列処理を行うことができる並列処理プロセッサおよび複数命令の並列実行方法を提供する。

【0004】

【従来の技術】図 7 は従来の長形式命令語の実行方法を示す。図 7 において、200 は命令語であって、N ビットの基本命令を M 個もつ固定長の長形式命令語であり、M=6 の場合を示す。図 7 では 1 タイミングで並列処理する基本命令 1、基本命令 2、基本命令 3 を含み、NOP1、NOP2、NOP3 (NOP は何も処理をしないことを指示する無操作命令である) を持たせ固定長の長形式命令語としたものである。

【0005】201 はデコーダであって、デコーダ 1、デコーダ 2、デコーダ 3、デコーダ 4、デコーダ 5、デコーダ 6 を持つものである。命令語 200 の各基本命令語に対応して設けられ、それぞれの基本命令をデコードするものである。

【0006】202は演算器であって、演算器1、演算器2、演算器3、演算器4、演算器5、演算器6を持ち、それぞれデコーダ1、デコーダ2、デコーダ3、デコーダ4、デコーダ5、デコーダ6でデコードされた命令を演算処理するものである。

【0007】203はレジスタであって、レジスタ1、レジスタ2、レジスタ3、レジスタ4、レジスタ5、レジスタ6を持ち、それぞれ演算器1、演算器2、演算器3、演算器4、演算器5、演算器6の演算結果を保持するものである。各レジスタは複数本の個別のレジスタを複数本備え、レジスタ空間を構成している。例えば、レジスタ1で1レジスタ空間を構成し、128ビットの個別レジスタを64本備える等である。

【0008】図7の構成の動作を説明する。命令語200のNビットの基本命令1、基本命令2、基本命令3、NOP1、NOP2、NOP3は同じタイミングでそれぞれ並列にデコーダ1、デコーダ2、デコーダ3、デコーダ4、デコーダ5、デコーダ6に入力され、それぞれ、演算器1、演算器2、演算器3、演算器4、演算器5、演算器6に入力される。

【0009】そして、基本命令1、基本命令2、基本命令3はそれぞれの演算器で演算処理され、演算結果はそれぞれレジスタ1、レジスタ2、レジスタ3に保持される。NOP1、NOP2、NOP3はそれぞれ処理を行わない指示であるので、演算されることはない。

【0010】図8は従来の長形式命令語の生成方法の説明図である。図8において、210は命令語生成プログラムであって、8個の基本命令語の固定長の長形式命令語を生成するプログラムである。

【0011】221は命令語生成プログラム210の第1行の記述で生成された命令語である。222は命令語生成プログラム210の第2行の記述で生成された命令語である。

【0012】223は命令語生成プログラム210の第3行の記述で生成された命令語である。224は命令語生成プログラム210の第4行の記述で生成された命令語である。

【0013】225は命令語生成プログラム210の第5行の記述で生成された命令語である。226は命令語生成プログラム210の第6行の記述で生成された命令語である。

【0014】227は命令語生成プログラム210の第7行の記述で生成された命令語である。命令語生成プログラム210の第1行の記述「add & sub」の記述により、第1番目の基本命令語が「add」、第2番目の基本命令語が「sub」、第3番目～第8番目の基本命令語がNOP（無操作命令）の長形式命令語221が生成される。

【0015】同様に、命令語生成プログラム210の第2行の記述「load & store」の記述によ

り、第1番目の基本命令語が「load」、第2番目の基本命令語が「store」、第3番目～第8番目の基本命令語がNOP（無操作命令）の長形式命令語222が生成される。

【0016】命令語223のように基本命令語が「load」命令のみで6個続くような場合にも、命令語生成プログラム210は第3行のように「load」命令を繰り返して記述する。

【0017】同様に、命令語生成プログラム210の第5行、第6行、第7行の記述により、長形式命令語224、225、226、227が生成される。

【0018】

【発明が解決しようとする課題】従来の長形式命令語による複数命令の並列処理方法は、固定長であったため、並列処理する基本命令数が所定の数に達しない場合には、NOP命令を付加して所定の長さにする必要があった。そのため、命令並列度の低いプログラムに対しては、NOPが多くなり、メモリの使用効率を悪化させ、キャッシュメモリのヒット率の低下、命令フェッチ機構の負荷の増大等をまねいていた。

【0019】一方、スーパースカラ方式では命令実行時の並列度に応じて、同時に実行される命令の長さが任意に変更されるので、効率の悪化は生じないが、並列度に応じて、必要なレジスタ数をどのようにして増加させるかが大きな課題とされている。

【0020】本発明は、長形式命令語を可変長として処理することができるようにして限られた数レジスタで効率良く並列処理することのできる並列処理プロセッサおよび並列実行方法を提供することを目的とする。

【0021】

【課題を解決するための手段】本発明は、複数の基本命令を一命令語としてまとめて複数の基本命令を並列処理する並列処理プロセッサにおいて、基本命令は同一タイミングで処理することの可否を指定する命令継続指定ビットを設けたものとし、複数の基本命令を基本命令毎に格納する命令バッファと、該命令バッファの基本命令をそれぞれに演算する演算部と、それぞれの演算器の演算結果を格納するレジスタ、該命令継続ビットの値を判別して基本命令のうち同一タイミングで処理を行ってはいない基本命令の演算部への入力を抑止する命令抑止部とを備え、命令抑止部は同一タイミングで処理をしてはいないことを指示された基本命令より後の基本命令の演算部への入力は抑止し、それ以前の同一タイミングで演算できる引き続き基本命令を一命令語とみなし、同一タイミングで演算部に入力して並列処理するようにした。

【0022】図1は本発明の基本構成を示す。図1(a)において、1は並列処理プロセッサである。

【0023】2は命令語であって、複数の基本命令を含むものである。図1は基本命令が6個で一命令を構成す

る場合を例として説明する。2' は命令バッファであって、命令語 2 の各基本命令を入力して保持するものである。3 は命令抑止部であって、抑止器 1, 抑止器 2, 抑止器 3, 抑止器 4, 抑止器 5, 抑止器 6, . . . , 抑止器 M を持つものである。各抑止器は命令語 2 のそれぞれに対応する位置の基本命令を命令バッファ 2' より入力し、命令語の各基本命令の持つ命令継続指定ビットに従って、同一タイミングで実行することのできない基本命令の実行を抑止するものである。命令継続指定ビットは例えば、0 であれば同一タイミングで実行するものとし、1 は同一タイミングで実行できる一連の基本命令の終了を示すものである。

【0024】4 はデコーダ部であって、各抑止器に対応するデコーダ 1, デコーダ 2, デコーダ 3, デコーダ 4, デコーダ 5, デコーダ 6, . . . , デコーダ M を持ち、それぞれに対応する抑止器の判定結果に応じて入力される命令語 2 の基本命令をデコードするものである。

【0025】5 は演算部であって、演算器 1, 演算器 2, 演算器 3, 演算器 4, 演算器 5, 演算器 6, . . . , 演算器 M を持ち、各演算器はそれぞれに対応するデコーダの演算結果を入力するものである。

【0026】6 はレジスタ部であって、レジスタ空間 1, レジスタ空間 2, レジスタ空間 3, レジスタ空間 4, レジスタ空間 5, レジスタ空間 6, . . . , レジスタ空間 M を持ち、各レジスタ空間はそれぞれに対応する演算器の演算結果を保持するものである。

【0027】命令語 2 において、10, 11 は命令継続指定ビットである。

【0028】

【作用】図 1 (a) の動作を説明する。命令抑止部 3 の各抑止器 (1 ~ M) はそれぞれに対応する位置の基本命令を入力する。各抑止器は、入力したそれぞれの基本命令の命令継続抑止ビットの値を判定する。例えば、値が 0 であれば同一タイミングで並列処理する命令であるとし、値が 1 であればその後続く基本命令はデコーダへの入力を抑止する (命令継続抑止ビットの値 1 の基本命令までデコーダに入力する)。

【0029】そこで、デコーダ部 4 は、それぞれのデコーダに入力された基本命令をデコードし、それぞれに接続される演算器 (1 ~ M) に基本命令を入力する。演算部 5 はそれぞれの演算器 (1 ~ M) において入力された命令を演算処理する。レジスタ部 6 はそれぞれに接続されている演算器の演算結果を対応するレジスタに格納する。

【0030】図 1 (b) は本発明で並列処理する命令語の長さと同じタイミングで並列処理できる基本命令数とレジスタ空間数の関係についての説明図である。図 1 (b) は、レジスタ空間数が M 個であるとする。また、命令語に含まれる基本命令語の長さは N ビットであり、命令語に含まれる基本命令数は P 個であるとする。

【0031】値 0 の命令継続ビットが引き続く基本命令列とその一連の命令継続ビットに続いて現れる値 1 の命令継続ビットの基本命令まで一命令語とみなし、最大 M 個 ($P=1, 2, \dots, M$) まで、同一タイミングで処理することができる。即ち、並列できる 1 命令語の長さは $P=1, 2, \dots, M$ の場合の M 組が可能である。

【0032】図 2 は本発明の動作例である。図 2 (a) は動作例であり、図 2 (b) はレジスタ空間の説明図である。図 2 (a) において、21 は命令語 1 であって、基本命令 1, 基本命令 2, 基本命令 3, 基本命令 4, 基本命令 5 を持つものである。基本命令 1 の命令継続ビットは 0, 基本命令 2 の命令継続ビットは 1 であって、基本命令 1 と基本命令 2 は前回に処理されたものである。

【0033】22 は命令語 2 であって、命令語 1 に続く命令であって、基本命令 1, 基本命令 2, 基本命令 3, 基本命令 4, 基本命令 5, 基本命令 6 を持つものである。命令語 1 (21) の基本命令 1 から基本命令 5 および命令語 2 (22) の基本命令 1 から基本命令 4 まで命令継続ビットは 0 であり、命令語 2 (22) の基本命令 5 で命令継続ビットが 1 になるので、命令語 1 (21) の基本命令 3 から基本命令 2 (22) の基本命令 5 までが今回の処理の対象になる。命令語 2 (22) の基本命令 6 より後の命令は次回に処理される。

【0034】29 は命令バッファであって、命令語の基本命令を入力して保持するものである。30 は命令抑止部であって、抑止器 1 ~ 抑止器 8 を備えるものである。

【0035】31 はデコーダ部であって、デコーダ 1 ~ デコーダ 8 を備えるものである。32 は演算部であって、演算器 1 ~ 演算器 8 を備えるものである。33 はレジスタ部であって、レジスタ空間 1 ~ レジスタ空間 8 を備えるものである。

【0036】図 2 (a) において、命令語 1 (21) の基本命令 2 までが、前回のタイミングで処理されている。今回のタイミングで命令抑止部 30 の各抑止器はそれぞれ対応する基本命令を入力する。抑止器 1 は命令語 1 (21) の基本命令 3, 抑止器 2 は基本命令 4, 抑止器 3 は基本命令語 5, および、抑止器 4 は次の命令語 2 (22) の基本命令 1, 抑止器 5 は基本命令 2, 抑止器 6 は基本命令 3, 抑止器 7 は基本命令 4, 抑止器 8 は基本命令 5 を入力する。そこで、それぞれの抑止器は命令継続ビットを判定し、命令継続ビットが 0 の引き続く基本命令および命令継続ビットが 1 になる命令語 2 (22) の基本命令 6 までの基本命令を抑止することなくそれぞれに接続されるデコーダに転送する。図 2 (a) は抑止されない基本命令数とレジスタ空間数は同じであるが、例えば、命令語 2 (22) の基本命令 4 の命令継続ビットが 1 であるとするれば、その基本命令 4 までがデコーダに転送される。

【0037】デコーダ部 31 の各デコーダはそれぞれに入力された基本命令をデコードし、演算部 32 の各演算

器はそれぞれの命令を演算処理し、レジスタ部33のレジスタ空間はそれぞれに接続される演算器の演算結果を保持する。

【0038】図2(b)はレジスタ空間の説明図である。40はレジスタ空間1であって、個別レジスタ45を複数個(例えば64個)もつものである。

【0039】41はレジスタ空間2である。42はレジスタ空間Mである。45は個別レジスタであって、例えば128ビットのレジスタである。

【0040】図2(b)は1レジスタ空間に例えば128ビットの個別レジスタを64本もつことを表している。なお、上記の説明において、命令バッファ中の命令継続ビットがすべて0の場合には、命令形式の例外が検出され、オペレーティングシステムに対して、命令の先頭アドレスが通知される。オペレーティングシステムは、この「ハードウェアの許容範囲を越える長い命令語」を適当な長さに分割して再実行する。あるいは命令バッファに保持した基本命令を順次シミュレートすることにより各基本命令語を実行する。

【0041】本発明によれば、固定長の長形式命令語であっても、可変長命令語として扱うことができる。そして、同じタイミングで並列処理できる基本命令が少ない場合には、次の命令語で同じタイミングで処理できる基本命令を並列処理できる。そのため、並列処理の実行速度が速くなるとともに、命令語を固定長にするために付加するNOP命令等が少なくなりメモリの利用効率を向上させることができる。

【0042】

【実施例】図3は本発明の実施例構成を示す。図3において、51は現命令語である。

【0043】A0は基本命令であって、命令継続指定ビットが0のものである。A1は基本命令であって、命令継続指定ビットが0のものである。A2は基本命令であって、命令継続指定ビットが0のものである。

【0044】A3は基本命令であって、命令継続指定ビットが0のものである。A4は基本命令であって、命令継続指定ビットが0のものである。A5は基本命令であって、命令継続指定ビットが1のものである。

【0045】52は次命令語である。B0は基本命令であって、命令継続指定ビットが0のものである。B1は基本命令であって、命令継続指定ビットが0のものである。

【0046】55は並列処理プロセッサである。58は命令バッファであって、8個のバッファ(60, 61, 62, 63, 64, 65, 66, 67)を備えるものである。図3は、それぞれのバッファに基本命令(A0), 基本命令(A1), 基本命令(A2), 基本命令(A3), 基本命令(A4), 基本命令(A5), 基本命令(B0), 基本命令(B1)を格納した状態を示す。

【0047】71は命令継続指定ビット検出器であって、バッファ60の基本命令(A0)の命令継続指定ビット(0)を入力して、その値を検出するものである。72は命令継続指定ビット検出器であって、バッファ61, 62の基本命令(A0, A1)の命令継続指定ビットを入力して、その値を検出するものである。

【0048】73は命令継続指定ビット検出器であって、バッファ60, 61, 62の基本命令(A0, A1, A2)の命令継続指定ビットを入力して、その値を検出するものである。

【0049】74は命令継続指定ビット検出器であって、バッファ60, 61, 62, 63の基本命令(A0, A1, A2, A3)の命令継続指定ビットを入力して、その値を検出するものである。

【0050】75は命令継続指定ビット検出器であって、バッファ60, 61, 62, 63, 64の基本命令(A0, A1, A2, A3, A4)の命令継続指定ビットを入力して、その値を検出するものである。

【0051】76は命令継続指定ビット検出器であって、バッファ60, 61, 62, 63, 64, 65の基本命令(A0, A1, A2, A3, A4, A5)の命令継続指定ビットを入力して、その値を検出するものである。

【0052】77は命令継続指定ビット検出器であって、バッファ60, 61, 62, 63, 64, 65, 66の基本命令(A0, A1, A2, A3, A4, A5, B0)の命令継続指定ビットを入力して、その値を検出するものである。

【0053】80は命令抑止部であって、抑止器81~87を備え、各命令継続指定ビット検出器(71~77)の検出値に応じて入力した基本命令の通過もしくは抑止を行うものである。

【0054】81は抑止器1であって、バッファ61の基本命令(A1)を入力し、命令継続指定ビット検出器71の検出値に応じて入力した基本命令(A1)の通過もしくは抑止を行うものである。

【0055】82は抑止器2であって、バッファ62の基本命令(A2)を入力し、命令継続指定ビット検出器72の検出値に応じて入力した基本命令(A2)の通過もしくは抑止を行うものである。

【0056】83は抑止器3であって、バッファ63の基本命令(A3)を入力し、命令継続指定ビット検出器73の検出値に応じて入力した基本命令(A3)の通過もしくは抑止を行うものである。

【0057】84は抑止器4であって、バッファ64の基本命令(A4)を入力し、命令継続指定ビット検出器74の検出値に応じて入力した基本命令(A4)の通過もしくは抑止を行うものである。

【0058】85は抑止器5であって、バッファ65の基本命令(A5)を入力し、命令継続指定ビット検出器

75の検出値に応じて入力した基本命令(A5)の通過もしくは抑止を行うものである。

【0059】86は抑止器6であって、バッファ66の基本命令(B0)を入力し、命令継続指定ビット検出器76の検出値に応じて入力した基本命令(B0)の通過もしくは抑止を行うものである。

【0060】87は抑止器7であって、バッファ67の基本命令(B1)を入力し、命令継続指定ビット検出器77の検出値に応じて入力した基本命令(B1)の通過もしくは抑止を行うものである。

【0061】90はデコーダ部であって、デコーダ1、デコーダ2、デコーダ3、デコーダ4、デコーダ5、デコーダ6、デコーダ7、デコーダ8を備えるものである。100は演算部であって、演算器1、演算器2、演算器3、演算器4、演算器5、演算器6、演算器7、演算器8を備えるものである。

【0062】110はレジスタ部であって、レジスタ空間1、レジスタ空間2、レジスタ空間3、レジスタ空間4、レジスタ空間5、レジスタ空間6、レジスタ空間7、レジスタ空間8を備えるものである。

【0063】図3の構成の動作を説明する。命令バッファ58の各バッファ(60, 61, 62, 63, 64, 65, 66, 67)はそれぞれ、基本命令A0, A1, A2, A3, A4, A5, B0, B1を入力する。

【0064】基本命令A0は直接デコーダ1に入力される。命令継続指定ビット検出器71は基本命令A0の命令継続指定ビット(0)を入力する。そして、抑止器は命令継続指定ビット検出器71の検出結果と基本命令A1を入力する。命令継続指定ビットが0なので抑止器81は基本命令A1を通過させてデコーダ2に入力する。

【0065】命令継続指定ビット検出器72は基本命令A0, A1の命令継続指定ビット(0)を入力する。そして、抑止器82は基本命令A2および命令継続指定ビット検出器72の検出結果を入力する。命令継続指定ビットが全て0なので抑止器82は基本命令A2を通過させてデコーダ3に入力する。

【0066】命令継続指定ビット検出器73は基本命令A0, A1, A2の命令継続指定ビット(0)を入力する。そして、抑止器83は基本命令A3と命令継続指定ビット検出器73の検出結果を入力する。命令継続指定ビットが全て0なので基本命令A3を通過させてデコーダ4に入力する。

【0067】命令継続指定ビット検出器74は基本命令A0, A1, A2, A3の命令継続指定ビット(0)を入力する。そして、抑止器84は基本命令A4と命令継続指定ビット検出器74の検出結果を入力する。抑止器84は命令継続指定ビットが全て0なので基本命令A4を通過させてデコーダ5に入力する。

【0068】命令継続指定ビット検出器75は基本命令A0, A1, A2, A3, A4の命令継続指定ビット

(0)を入力する。そして、抑止器85は基本命令A5と命令継続指定ビット検出器75の検出結果を入力する。抑止器85は命令継続指定ビットが全て0なので基本命令A5を通過させてデコーダ6に入力する。

【0069】命令継続指定ビット検出器76は基本命令A0, A1, A2, A3, A4の命令継続指定ビット(0)と基本命令A5の命令継続指定ビット(1)を入力する。そして、抑止器86は基本命令B0と命令継続指定ビット検出器76の検出結果を入力する。抑止器86は命令継続指定ビットに1があるので基本命令B0の通過を抑止する。

【0070】命令継続指定ビット検出器77は基本命令A0, A1, A2, A3, A4の命令継続指定ビット(0)と基本命令A5の命令継続指定ビット(1)と基本命令B0の命令継続指定ビット(0)を入力する。そして、抑止器87は基本命令B1と命令継続指定ビット検出器77の検出結果を入力する。抑止器87は命令継続指定ビットに1があるので基本命令B1の通過を抑止する。

【0071】デコーダ1、デコーダ2、デコーダ3、デコーダ4、デコーダ5、デコーダ6はそれぞれに入力された基本命令(A0, A1, A2, A3, A4, A5)をデコードする。各デコーダでデコードされた基本命令はそれぞれに接続される演算器に入力され、演算される。そして、演算結果はそれぞれの演算器(1~6)に接続されるレジスタ空間(1~6)に保持される。

【0072】図4は本発明の実施例1であって、レジスタ間でデータ転送を行う場合の構成を示す。図4において、120は長形式命令語であって、Nビットの基本命令語121がP個(P=1, 2, ..., M(最大レジスタ空間数))で構成され、それぞれに命令継続指定ビットを持つものである。

【0073】121は基本命令語である。130はデコーダ部であって、デコーダ1、デコーダ2、デコーダ3、デコーダ4、デコーダ5、デコーダ6により構成されるものである。

【0074】131は演算部であって、演算器1、演算器2、演算器3、演算器4、演算器5、演算器6により構成されるものである。132はレジスタ部であって、レジスタ空間1、レジスタ空間2、レジスタ空間3、レジスタ空間4、レジスタ空間5、レジスタ空間6により構成されるものである。

【0075】133はバッファレジスタであって、前段のレジスタ空間のバッファレジスタのデータを複写して保持し、演算器1の指示によりレジスタ空間1に入力するものである。

【0076】134はバッファレジスタであって、演算器1の指示によりレジスタ空間1のデータを受け取り、保持するものである。135はバッファレジスタであって、バッファレジスタ134のデータを複写して保持し、演算器2の指示によりレジスタ空間2に入力するも

のである。

【0077】136はバッファレジスタであって、演算器2の指示によりレジスタ空間2のデータを受け取り、保持するものである。137はバッファレジスタであって、バッファレジスタ136のデータを複写して保持し、演算器3の指示によりレジスタ空間3に入力するものである。

【0078】138はバッファレジスタであって、演算器3の指示によりレジスタ空間3のデータを受け取り、保持するものである。139はバッファレジスタであって、バッファレジスタ138のデータを複写して保持し、演算器4の指示によりレジスタ空間4に入力するものである。

【0079】140はバッファレジスタであって、演算器4の指示によりレジスタ空間4のデータを受け取り、保持するものである。141はバッファレジスタであって、バッファレジスタ140のデータを複写して保持し、演算器5の指示によりレジスタ空間5に入力するものである。

【0080】142はバッファレジスタであって、演算器5の指示によりレジスタ空間5のデータを受け取り、保持するものである。143はバッファレジスタであって、バッファレジスタ142のデータを複写して保持し、演算器6の指示によりレジスタ空間6に入力するものである。

【0081】144はバッファレジスタであって、演算器6の指示によりレジスタ空間6のデータを受け取り、保持するものである。バッファレジスタ144のデータは後段のバッファレジスタに複写される。

【0082】146は長形式命令語であって、その基本命令1はバッファレジスタ (a) 134にレジスタ空間1の内容を保持させるストア命令 (store Reg→a) を含むものである。

【0083】147は長形式命令語であって、その基本命令2はバッファレジスタ (b) 135の内容をレジスタ空間2にロードする命令 (Load b→Reg) を含むものである。

【0084】レジスタ空間の間でデータ転送を行う場合には、まずデータを送信するレジスタ空間に対応する基本命令に、次のような操作を行う。レジスタ空間1からレジスタ空間2に転送する場合を例として説明する。

【0085】まず、長形式命令語146の基本命令1にバッファレジスタ a (バッファレジスタ134) にレジスタ空間1の内容をストアする命令 (store Reg→a) をセットし (基本命令1へのセットは転送プログラムの記述に従ってコンパイラによりなされる)、バッファレジスタ (a) にレジスタ空間1の内容を保持させる。次にデータを受け取るレジスタ空間2に対応する基本命令2にバッファレジスタ b の内容をレジスタ空間2にロードする命令 (load Reg→b) をセットし

(基本命令2へのセットは、転送プログラムに従ってコンパイラによりなされる)、バッファレジスタ b の内容をレジスタ空間2にロードする。

【0086】図5は本発明の実施例2である。図5において、150はフェッチした命令語 (1) であり、基本命令となるものである。

【0087】151は命令拡張制御レジスタ設定の基本命令語であって、命令拡張制御レジスタ152に命令語における基本命令語の繰り返しパターンを表すマスクパターンを設定するものである。

【0088】152は命令拡張制御レジスタであって、基本命令1を繰り返し長形式命令語において、基本命令1のある位置を表すビットと基本命令1でない命令であることを識別するビットにより構成されるマスクパターンを持つものである。

【0089】153は命令語生成部であって、フェッチした基本命令語 (1) 150と命令拡張制御レジスタ152のマスクパターンにより命令語154を生成するものである。

【0090】154は実行される長形式の命令語である。本実施例2では、命令語の最大サイズをM (基本命令がM個) とした時、基本命令語の位置 (1, 2, ..., M) に1ビットずつ対応させたMビットの命令拡張制御レジスタ設定の基本命令語151を設ける。そして、命令語に繰り返される基本命令語1 (150) をフェッチした場合には、フェッチした命令1を基本命令とするか、あるいはNOPもしくはなにも命令をセットしないかを表すビットパターンを命令拡張制御レジスタにセットしておく。受け取った基本命令語が命令拡張制御レジスタ152のマスクパターンに従うものであるかないかは、例えば、フェッチした命令語が一命令語のみの場合には、マスクパターンに従って繰り返される基本命令であると判定する。そして、その命令1をフェッチすると、命令拡張制御レジスタ設定の基本命令語151を参照し、命令拡張制御レジスタ152のビット位置Kの値が0であれば位置Kに対応する基本命令語をNOPとする。パターンの途中から最後まで値0が続く場合には基本命令をなにもセットしなくて良い (図6参照)。また位置Kの値が1であれば基本命令を1と見なすことにより、長形式の命令語154を実行する。

【0091】図6は本発明の実施例2の命令語生成方法の説明図である。図6において、160は命令語生成プログラムである。

【0092】第1行「set len 0011111100」は命令拡張制御レジスタにマスクパターン「111111100」を設定する基本命令語である。第2行は基本命令語が「add & sub」の基本命令語「add」と基本命令語「sub」の2個の基本命令語により構成される命令語171を生成する記述である。

【0093】第3行「load & store」は基

本命令語「load」と基本命令語「store」の2個の基本命令語により構成される命令語172を生成する記述である。

【0094】第4行の「load」は基本命令語が「load」のみであり、それが6個繰り返される命令語173を生成する記述である。第4行の「load」は基本命令語が「load」のみであり、それが6個繰り返される命令語173を生成する記述である。

【0095】第5行の「add」は基本命令語が「add」のみであり、それが6個繰り返される命令語174を生成する記述である。第6行の「mult」は基本命令語が「mult」のみであり、それが6個繰り返される命令語175を生成する記述である。

【0096】第7行の「store」は基本命令語が「store」のみであり、それが6個繰り返される命令語176を生成する記述である。第8行「branch & nop」は基本命令語「branch」と基本命令語が「nop」の2個の基本命令語により構成される命令語を作成するものである。

【0097】図5を参照し、図6の本発明の実施例2の命令語生成方法について説明する。命令拡張制御レジスタ設定の基本命令語151（図6の命令語生成プログラム160の第1行の命令）を入力し、命令拡張制御レジスタにその命令で与えられているマスクパターン（11111100）を設定する（図5のマスクパターン（11011111）とはマスクパターンが異なる）。

【0098】第2行目の命令語のプログラムの記述「add & sub」は2基本命令の記述なので、マスクパターンに従う命令語生成はせずに、命令語171を生成する。

【0099】第3行目も同様にプログラムの記述「load & store」が2基本命令の記述なので、マスクパターンにはよらずに命令語172を生成する。第4行「load」の記述に従ってフェッチした命令語が1基本命令のみであるので、命令語生成部153は命令拡張制御レジスタ152を参照し、マスクパターンに従って基本命令語「load」を6個繰り返す命令語173を生成する。

【0100】第5行の記述に従って、同様にフェッチした命令語も1基本命令「add」のみであるので、命令拡張制御レジスタ152のマスクパターンに従って、命令語生成部153が命令語174を生成する。

【0101】第6行の記述に従ってフェッチした命令語も基本命令「mult」のみであるので、命令拡張制御レジスタ152のマスクパターンが参照されて命令語生成部153により命令語175が生成される。

【0102】第7行の記述に従って、フェッチした命令語「store」も1基本命令だけなので命令拡張制御レジスタ152のマスクパターンが参照されて命令語176が生成される。

【0103】第8行は「branch」命令のみで良いのであるが、基本命令語が「branch」命令1語であると、命令拡張制御レジスタが参照され「branch」命令を6個繰り返す命令語が生成されるので、それを防ぐために基本命令語に「nop」を付与して2個の基本命令の記述とし、命令語177を生成する。

【0104】なお、本発明では、図6の各命令語のように命令語の基本命令から後に基本命令が続かない場合にはにも基本命令を加える必要はない。本実施例の命令語生成プログラム160を図8の従来の命令語生成プログラム210と比較すると、従来の命令語生成プログラム210では合計で29個の基本命令語の記述を必要とするのに対して、本発明では合計で11個の基本命令の記述で良い。また、従来の命令語では、多数のNOP命令を付け加えて固定長の命令語とする必要があるのに対して、本発明では、図6のようにその必要がないので、メモリの使用効率が大幅に向上する。またキャッシュメモリ等におけるヒット率が高くなる。

【0105】

【発明の効果】本発明によれば、固定長の長形式命令語であっても、可変長命令語として扱うことができる。そして、同じタイミングで並列処理できる基本命令が少ない場合には、次の命令語で同じタイミングで処理できる基本命令を並列処理できる。そのため、並列処理の実行速度が速くなるとともに、命令語を固定長にするために付加するNOP命令等が少なくなりメモリの利用効率を大幅に向上させることができる。また、キャッシュメモリにおけるヒット率も高くなる。

【図面の簡単な説明】

【図1】本発明の基本構成を示す図である。

【図2】本発明の動作例を示す図である。

【図3】本発明の実施例構成を示す図である。

【図4】本発明の実施例1を示す図である。

【図5】本発明の実施例2を示す図である。

【図6】本発明の実施例2の命令語生成方法を示す図である。

【図7】従来の長形式命令語の実行方法を示す図である。

【図8】従来の長形式命令語の生成方法を示す図である。

【符号の説明】

1：並列処理プロセッサ

2：命令語

2'：命令バッファ

3：命令抑止部

4：デコーダ

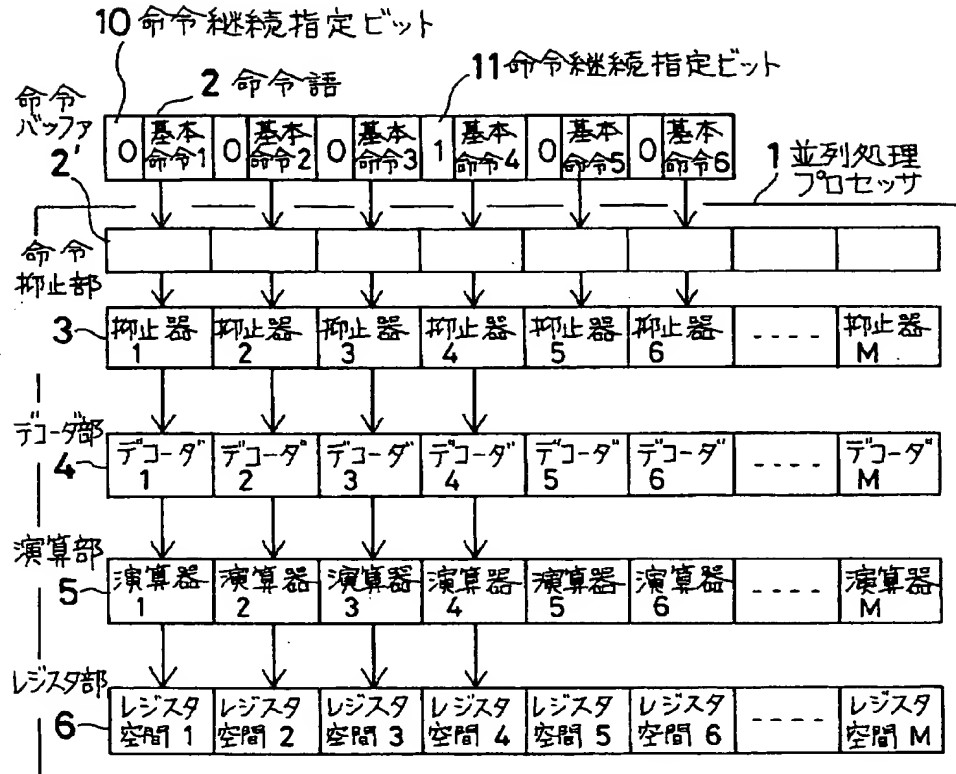
5：演算部

6：レジスタ部

10, 11：命令継続指定ビット

【図 1】

本発明の基本構成



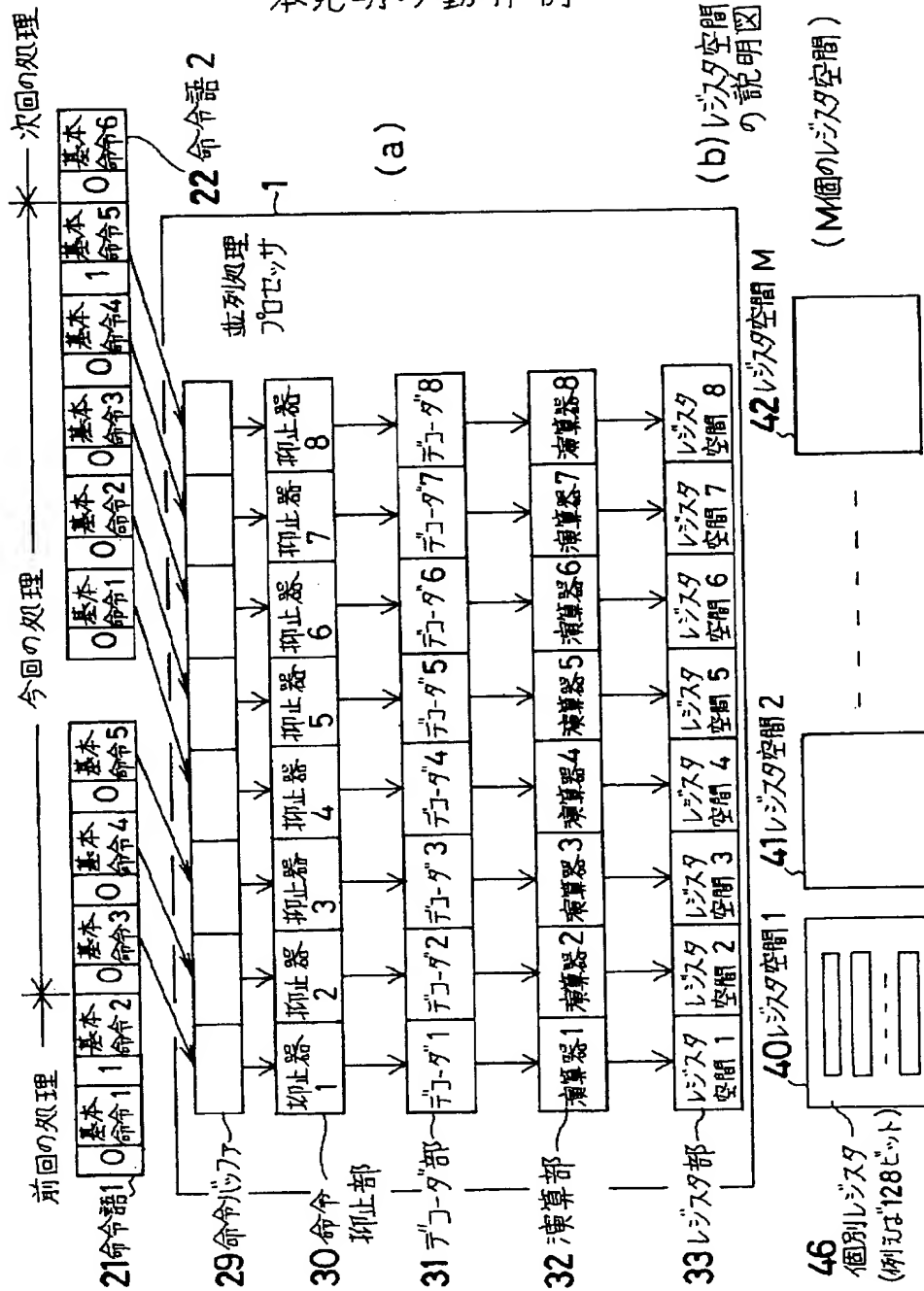
(a) 本発明の並列プロセッサの基本構成



(b) 本発明の命令実行方法

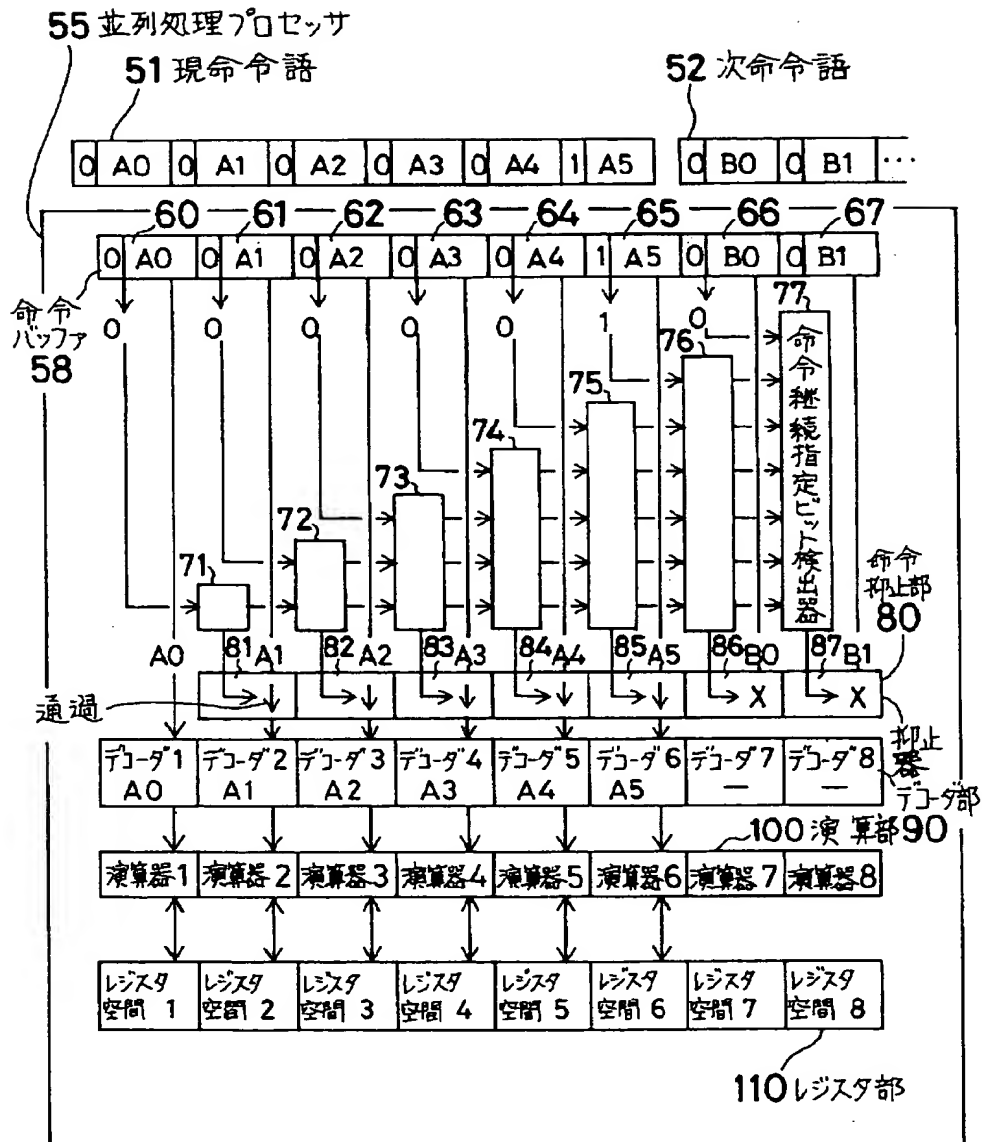
【図 2】

本発明の動作例



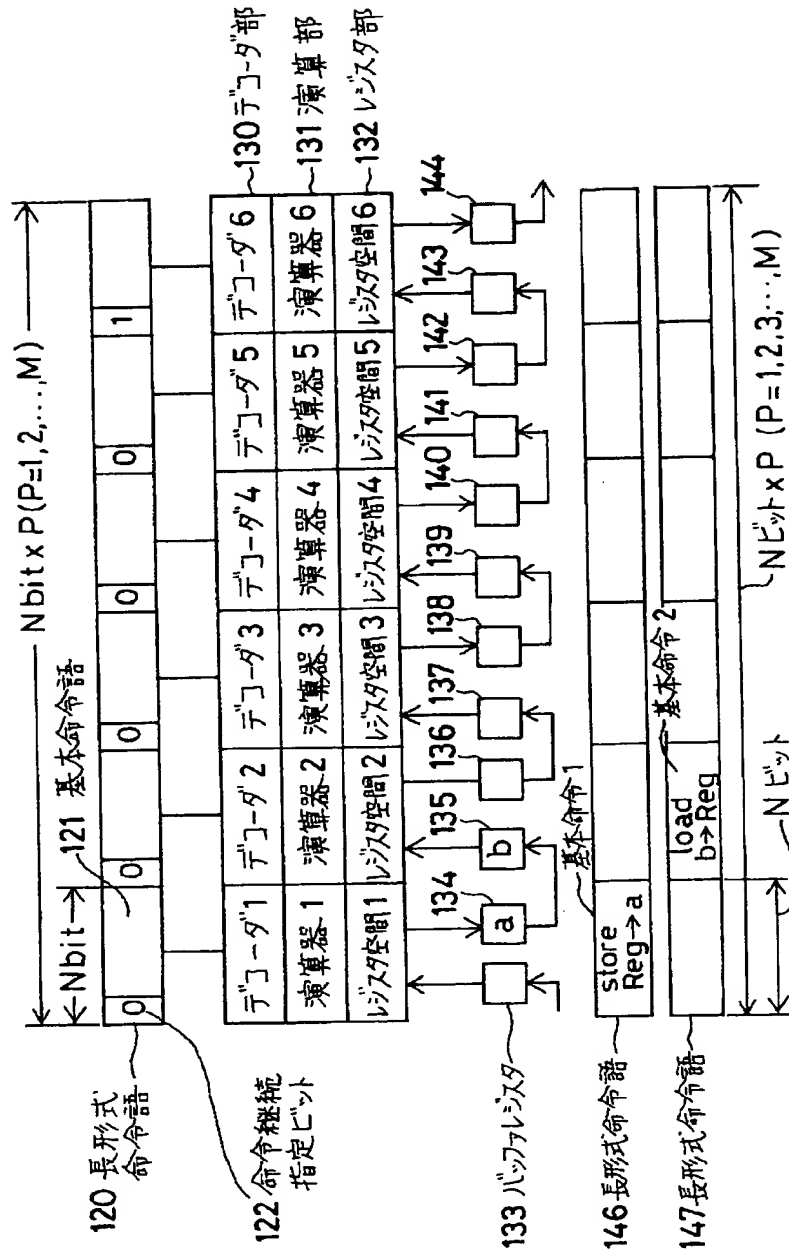
【図3】

本発明の実施例構成



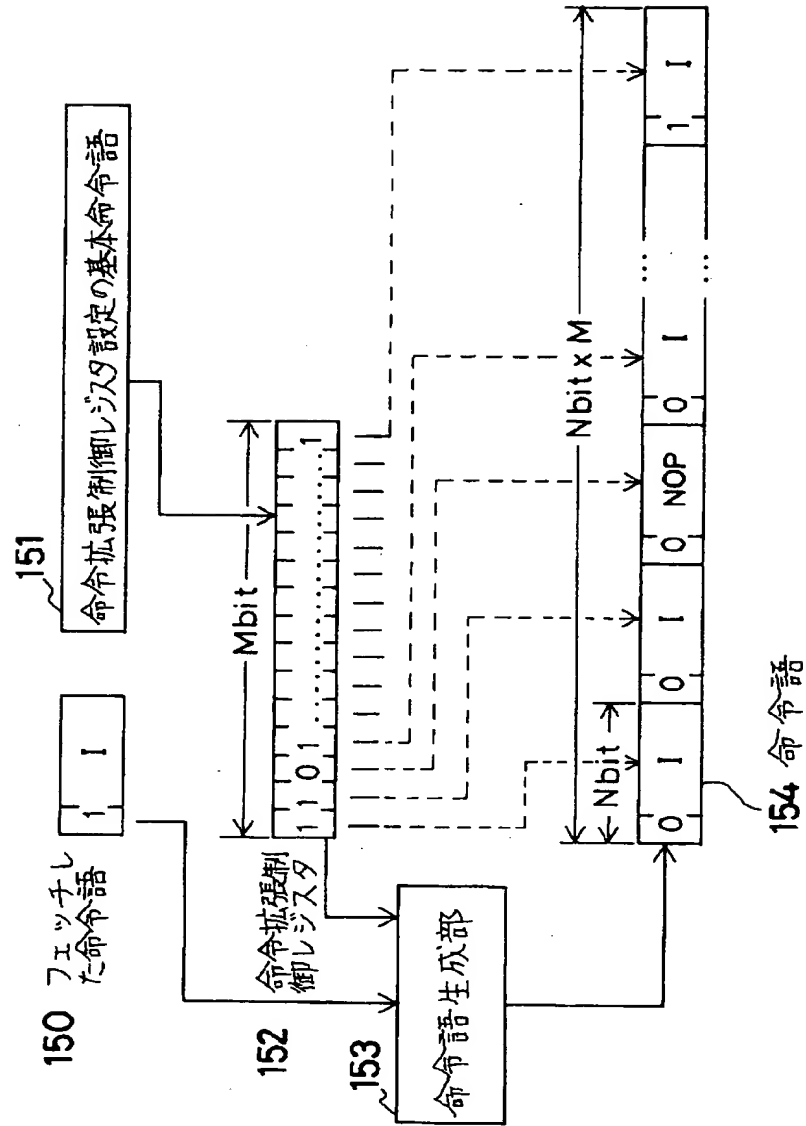
【図 4】

本発明の実施例 1



【図 5】

本発明の実施例 2



【図6】

本発明の実施例2の命令語生成方法

160 命令語生成プログラム

(第1行) setlen 0011111100 ;	基本命令語 X1
LOOP:(第2行) add & sub ;	基本命令語 X2
(第3行) load & store ;	基本命令語 X2
(第4行) load	基本命令語 X1
(第5行) add	基本命令語 X1
(第6行) mult	基本命令語 X1
(第7行) store	基本命令語 X1
(第8行) branch LOOP & nop ;	基本命令語 X2

(合計11個)

152 命令拡張制御レジスタ

1	1	1	1	1	1	0	0
---	---	---	---	---	---	---	---

171 (第2行)

add	sub						
-----	-----	--	--	--	--	--	--

172 (第3行)

load	store						
------	-------	--	--	--	--	--	--

173 (第4行)

load	load	load	load	load	load		
------	------	------	------	------	------	--	--

174 (第5行)

add	add	add	add	add	add		
-----	-----	-----	-----	-----	-----	--	--

175 (第6行)

mult	mult	mult	mult	mult	mult		
------	------	------	------	------	------	--	--

176 (第7行)

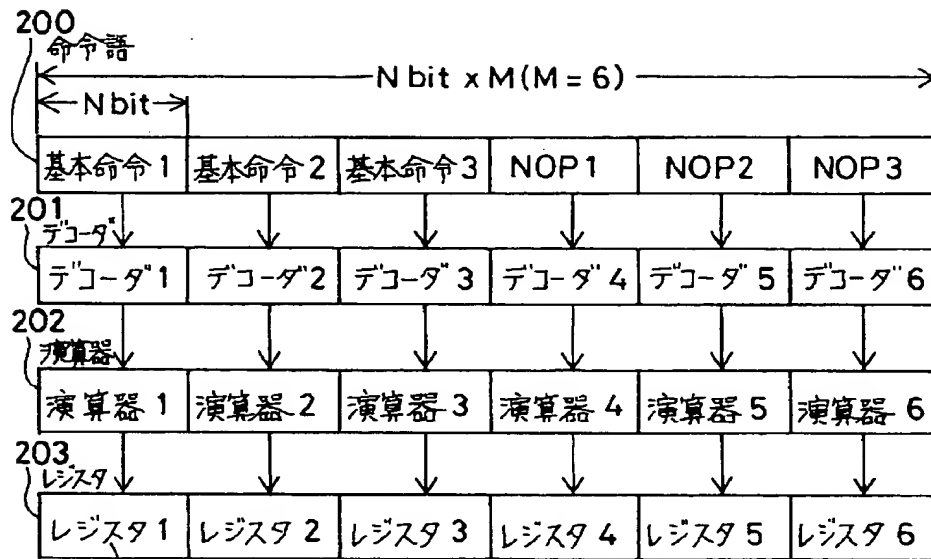
store	store	store	store	store	store		
-------	-------	-------	-------	-------	-------	--	--

177 (第8行)

branch	nop						
--------	-----	--	--	--	--	--	--

【図7】

従来の長形式命令語の実行方法



レジスタ空間 (複数本の個別レジスタを含む)

【図 8】

従来の長形式命令語の生成方法

210 命令語生成プログラム

LOOP: (1行) add	& sub;	-----	(基本命令語 X 2)
(2行) load	& store;	-----	(基本命令語 X 2)
(3行) load	& load & load & load & load & load	-----	(基本命令語 X 6)
(4行) add	& add & add & add & add & add	-----	(基本命令語 X 6)
(5行) mult	& mult & mult & mult & mult & mult	-----	(基本命令語 X 6)
(6行) store	& store & store & store & store & store	-----	(基本命令語 X 6)
(7行) branch	LOOP;	-----	(基本命令語 X 1)

(合計24個)

221	(1行)	add	sub	nop	nop	nop	nop	nop	nop
222	(2行)	load	store	nop	nop	nop	nop	nop	nop
223	(3行)	load	load	load	load	load	load	nop	nop
224	(4行)	add	add	add	add	add	add	nop	nop
225	(5行)	mult	mult	mult	mult	mult	mult	nop	nop
226	(6行)	store	store	store	store	store	store	nop	nop
227	(7行)	branch	nop	nop	nop	nop	nop	nop	nop